流媒体服务系统中一种基于数据预取的缓存策略

巫旭敏 殷保群 黄 静 郭 东
(中国科学技术大学网络传播系统与控制联合实验室网络传播系统与控制安徽省重点实验室 合肥 230027)

摘 要：具有 VCR 功能的流媒体服务系统由于请求的随机性会影响用户的点播体验，该文结合数据预取机制以及基于分段的缓存策略计算出用户点播延迟的期望，给出一个较优的缓存管理策略，并通过在线计算逼近最优解，同时在缓存已知的情况下，给出相应的数据预取算法，利用缓存和预取两种数据获取方法的相互协作减小客户端点播延迟，提高缓存效率。仿真结果证实了所提算法的有效性。

关键词：多媒体通信；预取机制；缓存策略；服务质量
中图分类号：TN919.85 **文献标识码**： A **文章编号**：1009-5896(2010)10-2440-06
DOI: 10.3724/SP.J.1146.2009.01333

A Prefetching-based Caching Policy in Streaming Service Systems

Wu Xu-min Yin Bao-qun Huang Jing Guo Dong
(*Joint Lab of Network Communication System and Control*, *Anhui Key Lab of Network Communication System and Control*, *University of Science and Technology of China*, *Hefei* 230027, *China*)

Abstract: Customers can not get high QoS from the streaming service systems with VCR operation because of the random requests. This paper derives the expectation of the demanding delay with the methods of prefetching and segment-based caching. A near-optimal policy of cache management is given, and the solution can approximate to the optimal one by computing online. The prefetching algorithm is given in the paper. The algorithm can reduce the delay of demands from clients with cooperation of caching and prefetching for improving the efficiency of cache. Simulation results show the effectiveness of proposed algorithm.

Key words: Multimedia communication; Prefetching scheme; Caching policy; Quality of Service (QoS)

1 引言

随着计算机网络发展，流媒体服务系统在应用上逐渐成熟。流媒体服务系统具有视频点播(Video On Demand, VOD)特性和实时性，一般使用缓存进行服务[1,2]由于存储容量有限，此类系统通常采用部分缓存[3–12]。这就需要适当的策略管理缓存。在内存管理中 LRU (Least Recently Used) 算法置换最近最少使用的数据，但流媒体服务系统受各节点间网络带宽限制以及存在大量不同内容同时被访问的现象，所以 LRU 对 QoS 的提高有限。基于热度的策略[5–10]通过缓存热度高的数据减小请求延迟和系统负载。热度是指各影片或影片片段被访问次数在总数据请求次数中所占的比率。文献[5]根据优化目标把多媒体文件分割成 3 部分分别存储在缓存、客户端以及中心服务器。文献[6]探讨了两种典型的分段策略：定长分段和指数分段。定长分段由于各段大小相同易于管理[7–12]。文献[5,7]将缓存管理归纳为动态规划，并利用贪心算法求解以降低计算复杂度。文献[8]针对流媒体系统的命中率、点播延迟以及网络抖动等性能进行研究，并给出基于多目标规划的解决方案。文献[9]进一步考虑了影片码率对存储策略的影响。

基于热度的缓存策略通过缓存访问频繁的数据减少本地节点向其它节点请求的数据量以及点播延迟。由于热度是通过统计给出的而未考虑用户的 VCR(Video Cassette Recording)行为[11,12]，当用户随机请求影片内容时，若请求数据段未被缓存，用户会获得较大的延迟。在 VOD 系统中 VCR 表示客户端对节目的快进、快退、暂停以及随机访问等交互操作。文献[11]采取折衷的策略，若请求数据未存储到本地就选择缓存中在播放时间上和该数据较接近的内容进行服务，在减小延迟的同时用户的请求会得到偏移。文献[12]针对客户端的 VCR 行为，研究 P2P 系统中的数据预取以及客户端的缓存管理。

在流媒体服务系统中，为了减小服务延迟以及系统负载，主要研究思路是利用有限的存储空间结合影片热度提高缓存效率，而实际上系统是通过缓存以及数据预取两种机制获得数据提供服务的，而

2009-10-15 收到，2010-04-13 改回
国家 863 计划项目(2008AA01A317)，国家自然科学基金(60935001)和安徽高校省级自然科学研究重点项目(KJ2009A152)资助课题
通信作者：巫旭敏 xuminwu@mail.ustc.edu.cn

当前大部分研究并没有考虑到数据预取对缓存效率的影响。本文的创新之处在于结合数据预取综合考虑流媒体服务系统中数据的起始延迟以及在服务过程中抖动所引起的延迟，计算出该延迟的期望，利用贪心算法给出缓存管理的次优解，并通过在线优化的方法提高缓存效率。

2　预取机制

本文采用 P2P-CDN (Content Distribution Network)的混合结构(图 1)，各缓存节点和中心服务器形成 P2P 网络。影片采用定长分段，中心服务器存储所有数据，缓存节点部分存储。当请求到达时，若本地缓存有客户请求的数据，本地缓存直接进行服务，否则，本地缓存向其它缓存或中心服务器请求数据，再对客户端提供服务。通常本地缓存和用户之间的传输延迟小，其数据传输速率大于影片码率，而缓存节点之间以及缓存到中心服务器之间的传输延迟较大，其数据传输速率小于影片码率。

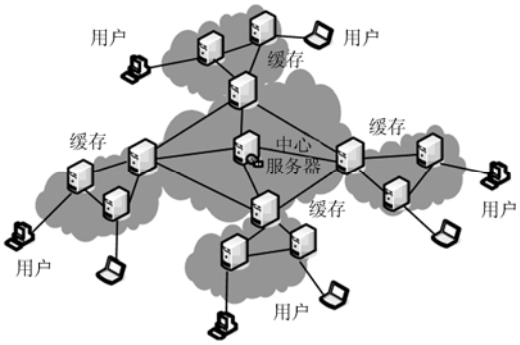

图 1 P2P-CDN 结构的流媒服务系统

假设有 N 部影片，其中第 v 部影片有 S_v 个数据段，其码率恒定为 r_v，数据段播放时间为 T_v，大小为 Seg，$\text{Seg} = r_v \cdot T_v$。若用户当前观看影片 v 的第 i 段，为了减小延迟，本地缓存应向其它节点请求未缓存到本地的数据，并决定需要预取的数据段以及进行带宽分配。假设点播段 i 时刻为 t_i，结束播放时刻为 $t_i + T_v$，当开始播放段 i 时，由于上一个预取周期的数据下载可能未完成，需要时间 θ_i 下载剩余数据，即开始预取时刻为 $t_i + \theta_i$，θ_i 称为预取时间间隔。用户结束段 i 播放后请求数据段用 j 表示，用 $t_i + T_v$ 表示用户请求段 j 的时刻。t_j 表示段 j 开始播放的时刻，则段 j 的请求时延 $\tau_j = t_j - (t_i + T_v)$，$\tau_j \geq 0$。预取过程如图 2 所示，播放段 i 时的预取从时间 $t_i + \theta_i$ 开始，在 $t_j + \theta_j$ 结束。

3　预取和缓存策略

记 $f_v(x,y)$，$x,y \in \{1,2,\cdots,S_v\}$，表示用户观看影片 v 从段 x 跳转到段 y 的统计次数，当 $x = y$ 时表示重播当前数据段，可以估计出用户观看影片 v 时从段 x 跳转到段 y 的概率

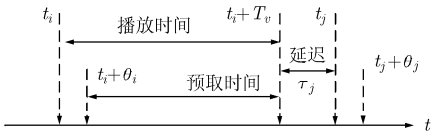

图 2 播放数据段 i 时的预取过程

$$PS_v(x,y) = \frac{f_v(x,y)}{\sum_{m=1}^{S_v}\sum_{n=1}^{S_v} f_v(m,n)} \tag{1}$$

若已知用户当前点播段 α，用户下一段点播段 y 的条件概率为

$$PS_v(y \mid \alpha) = \frac{PS_v(\alpha,y)}{\sum_{n=1}^{S_v} PS_v(\alpha,n)} \tag{2}$$

记影片 v 的所有数据段集合为 C^v，时刻 t 影片 v 存储在本地的数据段集合为 R_t^v，未存储在本地的数据段集合为 U_t^v，$C^v = U_t^v + R_t^v$。若影片 v 段 i 未缓存到本地，记 o_i^v 为系统中所有节点对该数据段的总上传带宽。b_d 为本地缓存的下载带宽，满足 $b_d \geq o_i^v$，即本地缓存下载带宽不小于任一数据段的上传带宽。r_k^v 为预取影片 v 段 k 的带宽，$r_k^v \leq o_k^v \leq b_d$。用户点播影片 v 段 i 后点播段 j，若段 j 未缓存到本地，当满足 $r_j^v \geq \text{Seg}/(T_v - \theta_i^v)$ 时，客户端没有延迟。记 $B_k^v = \text{Seg}/(T_v - \theta_i^v)$，若 $r_k^v > B_k^v$ 会降低下载带宽利用率，所以 $r_k^v \leq B_k^v$。当用户结束影片 v 段 i 播放后，此时已知下一个请求的数据段为 j，若段 j 未下载完，为了尽快获得需要的数据段本地缓存以系统上传带宽 o_j^v 进行下载。用 τ 表示客户端请求数据段的延迟，当 t 时刻客户正在观看影片 v 的段 i 时，下一个数据段延迟的期望为[12]

$$\begin{aligned} E\{\tau \mid v,i\} &= \sum_{k\in U_t^v} \frac{PS_v(k \mid i)[\text{Seg} - r_k^v(T_v - \theta_i^v)]}{o_k^v} \\ &= \sum_{k\in C^v} \frac{PS_v(k \mid i)\text{Seg}}{o_k^v} - \sum_{k\in R_t^v} \frac{PS_v(k \mid i)\text{Seg}}{o_k^v} \\ &\quad - \sum_{k\in U_t^v} \frac{PS_v(k \mid i) r_k^v (T_v - \theta_i^v)}{o_k^v} \end{aligned} \tag{3}$$

式(3)被分为 3 部分，其中第 1 部分表示没有缓存和预取机制的延迟期望，第 2 部分表示缓存机制所减小的延迟期望，第 3 部分表示预取机制所减小的延迟期望。定义 P_i^v 表示已知用户访问影片 v 时请求数据段为 i 的概率，$f_v(x,y)$ 中约定 $x = 0$ 表示用户从第 y 段开始请求影片 v，$y = 0$ 表示用户点播完段 x 后结束影片 v 的访问，即 $x,y \in \{0,1,2,\cdots,S_v\}$，有

$$P_i^v = \frac{\sum_{j=0}^{S_v} f_v(j,i)}{\sum_{j=0}^{S_v}\sum_{k=1}^{S_v} f_v(j,k)}, \quad i \in \{1,2,\cdots,S_v\} \tag{4}$$

通过式(3)可得

$$E\{\tau \mid v\} = \sum_{i=1}^{S_v} P_i^v E\{\tau \mid v,i\} \tag{5}$$

记 P^v 表示影片 v 的热度，它服从 Zipf 分布[7]，有系统延迟期望为

$$\begin{aligned} E\{\tau\} = \sum_{v=1}^{N} P^v E\{\tau \mid v\} &= \sum_{v=1}^{N}\sum_{i=1}^{S_v}\sum_{k\in C^v} \frac{P^v P_i^v PS_v(k \mid i)\mathrm{Seg}}{o_k^v} \\ &- \sum_{v=1}^{N}\sum_{i=1}^{S_v}\sum_{k\in R_t^v} \frac{P^v P_i^v PS_v(k \mid i)\mathrm{Seg}}{o_k^v} \\ &- \sum_{v=1}^{N}\sum_{i=1}^{S_v}\sum_{k\in U_t^v} \frac{P^v P_i^v PS_v(k \mid i) r_k^v (T_v - \theta_i^v)}{o_k^v} \end{aligned} \tag{6}$$

其中可以利用 $f_v(x,y)$ 计算出

$$P^v = \frac{\sum_{i=1}^{S_v} f_v(0,i)}{\sum_{m=1}^{N}\sum_{i=1}^{S_v} f_m(0,i)} \tag{7}$$

要使用户请求数据段的延迟期望最小，则应满足式(8)

$$\begin{aligned} \min E\{\tau\} = &\sum_{v=1}^{N}\sum_{i=1}^{S_v}\sum_{k\in C^v} \frac{P^v P_i^v PS_v(k \mid i)\mathrm{Seg}}{o_k^v} \\ &- \sum_{v=1}^{N}\sum_{i=1}^{S_v}\sum_{k\in R_t^v} \frac{P^v P_i^v PS_v(k \mid i)\mathrm{Seg}}{o_k^v} \\ &- \sum_{v=1}^{N}\sum_{i=1}^{S_v}\sum_{k\in U_t^v} \frac{P^v P_i^v PS_v(k \mid i) r_k^v (T_v - \theta_i^v)}{o_k^v} \end{aligned} \tag{8}$$

$$\text{s.t.} \quad 0 \le r_k^v \le \min\{o_k^v, B_k^v\} \tag{9}$$

$$\sum_{v=1}^{N}\sum_{k\in U_t^v} r_k^v \le b_d \tag{10}$$

$$\sum_{v=1}^{N}\sum_{k\in R_t^v} \mathrm{Seg} \le \mathrm{Stor} \tag{11}$$

Stor 表示本地缓存的大小，即式(11)表示缓存到本地的数据段受存储容量限制。R_t^v 表示影片 v ($v \in \{1,2,\cdots,N\}$)需要缓存的数据段以及 r_k^v ($k \in U_t^v$)表示影片未缓存的数据段的预取带宽。$f_v(x,y)$ 为一段时间内统计的结果，o_k^v, θ_i^v 也可以以通过一段时间的统计给出其均值，系统通过周期性地更改这些参数的信息进行缓存调整。定义 β 因子为

$$\beta_k^v = \sum_{i=1}^{S_v} \frac{P^v P_i^v PS_v(k \mid i)}{o_k^v} \tag{12}$$

γ 因子为

$$\gamma_k^v = \sum_{i=1}^{S_v} \frac{P^v P_i^v PS_v(k \mid i)(T_v - \theta_i^v)}{o_k^v} \tag{13}$$

式(8)可以等价为

$$\max \sum_{v=1}^{N}\sum_{k\in R_t^v} \beta_k^v \mathrm{Seg} + \sum_{v=1}^{N}\sum_{k\in U_t^v} \gamma_k^v r_k^v \tag{14}$$

式(14)第 1 项表示缓存减小的延迟期望，第 2 项表示预取减小的延迟期望。考虑已知 R_t^v 的情况求第 2 项，此时式(14)为有约束的线性规划，表示已知存储的情况下进行数据预取，有

$$\max \sum_{v=1}^{N}\sum_{k\in U_t^v} \gamma_k^v r_k^v \tag{15}$$

式(15)可用贪心算法求解，即优先预取 γ_k^v 值大的数据段。易证贪心算法的解为最优解。

由于 $0 \le r_k^v \le \min\{o_k^v, B_k^v\}$，而 $B_k^v = \mathrm{Seg}/(T_v - \theta_k^v)$，故有 $r_k^v(T_v - \theta_k^v) \le \mathrm{Seg}$，由式(6)可得同一数据段通过预取减小的延迟大于等于通过缓存减小的延迟。记 $R_t = \bigcup_{v=1}^{N} R_t^v$，表示 t 时刻缓存影片数据段的集合。由式(14)得，当 R_t 已知时，式(15)存在最优解。由于缓存容量有限，必然存在式(11)的约束。理论上式(14)可通过动态规划求最优解：首先求出不同 R_t 情况下的局部最优解，通过比较不同 R_t 情况下的局部最优解从而得出全局最优解。但由于动态规划计算复杂，通常利用贪心算法求式(14)次优解：将所有数据段按照 β 从大到小排序，存储 β 值较大的数据块，然后按照 γ 的大小预取未缓存数据块。由此可见，γ 因子为预取带宽分配权值，未被缓存且 γ 值大的数据段会获得较大的预取带宽分配。在贪心算法中，β 因子为缓存分配权值，β 值大的数据段会得到优先的缓存。

记 i^v 表示影片 v 的段 i，D_t 表示按照式(15)的贪心算法求解出的预取带宽不为 0 的数据段组成的集合，有 $D_t \subseteq U_t$，$U_t = \bigcup_{v=1}^{N} U_t^v$，考虑 $i_1^{v_1} \in R_t$，$i_2^{v_2} \notin R_t$ 的情况，若用 $i_2^{v_2}$ 置换缓存中的 $i_1^{v_1}$，记置换后缓存数据段，未缓存数据段以及预取数据段分别为 $R_{t'}, U_{t'}, D_{t'}$，其中满足 $i_1^{v_1} \notin R_{t'}$，$i_2^{v_2} \in R_{t'}$，$R_t \cup R_{t'} = R_t \cup \{i_2^{v_2}\} = R_{t'} \cup \{i_1^{v_1}\}$，置换缓存前后的请求数据段延迟的期望分别为 $E\{\tau\}$ 和 $E\{\tau'\}$，记 $\Delta E\{\tau\} = E\{\tau\} - E\{\tau'\}$，有以下 3 种情况：

情况 1 若 $i_1^{v_1} \notin D_{t'}, i_2^{v_2} \notin D_t$，显然有 $D_t = D_{t'}$，故若 $\beta_{i_2}^{v_2} < \beta_{i_1}^{v_1}$，置换前后延迟的期望之差 $\Delta E\{\tau\} = \beta_{i_2}^{v_2}\mathrm{Seg} - \beta_{i_1}^{v_1}\mathrm{Seg} < 0$；

情况 2 若 $i_1^{v_1} \in D_{t'}$，$i_2^{v_2} \notin D_t$，$\Delta E\{\tau\} = \beta_{i_2}^{v_2}\mathrm{Seg} - \beta_{i_1}^{v_1}\mathrm{Seg} + \gamma_{i_1}^{v_1} r_{i_1}^{v_1} - \gamma' r_{i_1}^{v_1}$，其中 γ' 为由于分配了 $r_{i_1}^{v_1}$ 的带宽给段 $i_1^{v_1}$ 而导致 D_t 部分未被预取数据的平均 γ 因子，式(15)是由贪心算法选择 γ 值较大的数据段进行预取，故有 $\gamma_{i_1}^{v_1} > \gamma'$，特别若有 $\beta_{i_2}^{v_2} \ge \beta_{i_1}^{v_1}$，$\Delta E\{\tau\} > 0$，即延迟 τ 的期望会减小；

情况 3　若 $i_1^{v_1}\in D_{t'}$, $i_2^{v_2}\in D_t$, $\Delta E\{\tau\}=\beta_{i_2}^{v_2}\mathrm{Seg}-\beta_{i_1}^{v_1}\mathrm{Seg}+\gamma_{i_1}^{v_1}r_{i_1}^{v_1}-\gamma_{i_2}^{v_2}r_{i_2}^{v_2}+\gamma''(r_{i_2}^{v_2}-r_{i_1}^{v_1})$，其中 γ'' 为 $i_2^{v_2}$ 置换 $i_1^{v_1}$ 前后引起的预取带宽的变化的 γ 因子，由式(15)所决定，$r_{i_2}^{v_2}>r_{i_1}^{v_1}$ 时 γ'' 为 $D_{t'}$ 中的较小值，$r_{i_2}^{v_2}<r_{i_1}^{v_1}$ 时为 D_t 中的较小值。只要选定初始解，可以通过(1),(2),(3)在线调整缓存内容，从而使得式(14)得到更好的解，算法步骤如下：

(1)选择 β 值较大的数据段缓存，确定 R_t，然后在 U_t 中选择 γ 值较大的数据段，确定 D_t；

(2)将 R_t 中的数据段按照 β 值从小到大排列索引，记 $\eta=2$ 为一个索引序号的初始值；

(3)当预取一个数据段时，分别按照情况 1，情况 2，情况 3 和 R_t 中索引为 1 的数据段进行比较，若索引为 1 的数据段不被替换，依次和索引为 $\eta,\eta+1,\cdots$ 的数据段进行比较，直到有数据段被预取数据段替换，记该数据段索引为 k，有 $\eta=k+1$，原索引为 $1,\cdots,k-1$ 的数据段索引号依次加 1，缓存的预取的数据段索引记为 1，在线继续进行步骤(3)。

由于替换的过程是按照索引号从小到大查找第 1 个满足替换条件的数据段，所以可以确定不能替换索引为 1 的数据段则一定不能替换索引为 $2,\cdots,\eta-1$ 的数据段。情况 2 中由于 $\gamma_{i_1}^{v_1}>\gamma'$，而 γ' 又必定大于等于 D_t 中最小的 γ 因子，故当 γ' 取 D_t 中最小的 γ 值时，令 $\Delta E\{\tau\}=0$，此时求的 $\beta_{i_1}^{v_1}$ 为比较的上限，即当 $i_1^{v_1}$ 的 β 因子高于此值时就可以终止比较，段 $i_2^{v_2}$ 不被存储在缓存中，类似可以求出情况 3 中的比较上限。

在 R_t 确定的情况下，可以按照式(15)利用贪心算法进行求解。式(15)是对一段时间进行统计的结果，而实际预取需要根据客户端实时请求的状况进行预取带宽分配，所以实际的预取策略需要将式(15)中的基于统计的参数用实时值替换进行求解。

4　仿真

考虑 3 部影片，热度服从 Zipf 分布，影片按热度从高到低排序，影片 v 的热度

$$P^v=1/v^{1-\mu}\Big/\sum_{v=1}^{N}(1/v^{1-\mu}) \tag{16}$$

仿真取 $\mu=0.271$ [7]，$N=3$。影片分别有 12， 13 和 11 个数据段，码率取为 500 kbps，数据段播放时间为 30 s，数据段为 15000 kb，缓存容量取为总数据量的 1/3 即缓存 12 个数据段，下载带宽 $b_d=6000$ kbps，各影片的上传带宽需高于影片的码率分别取为 500~1500 kbps 之间的随机数。客户端请求服从 Poisson 分布，请求的时间间隔服从参数为 60 s 的指数分布，请求数取 50000 次。客户首次请求首段的概率取 0.7~0.9 之间的随机数，首次请求其他数据段的概率服从均匀分布，连续请求即客户观看完影片段 i 接着观看影片段 $i+1$ 的概率取 0.4~0.6 之间的随机数，观看段 i 后请求其他数据段的概率服从均匀分布。对于式(8)中的 θ_i^v 其初始值取 0，再通过仿真系统运行一段时间统计给出其均值。算法 1 采用基于热度的缓存算法，即缓存热度大的数据段，当用户点播当前段时开始顺序预取其下一数据段。算法 2 采取本文中所描述的基于数据预取的策略。仿真后分别统计用户点播一段时间发生延迟的数据段总数以及发生延迟数据段的平均延迟。

图 3 为下载带宽分别取 4000 kbps, 6000 kbps 以及 8000 kbps 的延迟数据段段数和平均延迟。从图 3(a)可得，在相同时间点，基于预取算法的延迟数据段要少于基于热度算法的延迟数据段，在 4000 kbps 的时候两者发生延迟的数据段段数的差较小，这是因为带宽较小的时候，由于仿真中用户连续点播的概率相对访问其他数据段的概率大，所以采用顺序预取的方式能满足用户连续点播的请求，当带宽较大，预取算法能够更好地利用网络带宽进行数据预取，减小数据发生延迟的概率。带宽增大，顺序预取发生延迟的数据段段数变化不大，而预取算法随着带宽增大其发生延迟的数据段段数明显下降，这说明预取算法更能充分利用网络带宽提高流媒体系统的 QoS。图 3(b)表示预取算法的平均延迟要小于基于热度的算法。

当连续请求概率较大时，用户倾向于顺序播放的点播行为，此时基于热度的算法要好于基于预取的算法，图 4(a)中当连续请求概率取到 0.6~0.8 之间时，基于热度算法发生延迟的数据块要小于预取算法，但图 4(b)显示预取算法数据块的平均延迟较热度算法小。当连续请求概率取到 0.4~0.6 和 0.2~0.4 之间时，预取算法具有更大的优势。随着连续请求概率的减小，基于热度的算法其发生延迟的数据块数增加，延迟的平均值也在增加，这说明基于热度的预取算法不适合于用户随机点播行为明显的视频服务系统。

从仿真结果看，基于数据预取的缓存机制能够减小数据段的延迟，并且更能充分地利用网络带宽提高 QoS，当用户进行 VCR 操作时能够保障用户的点播体验。而基于热度的算法更适用于网络带宽较小以及用户顺序点播行为较明显的系统。记 $Z=\sum_{i=1}^{N}S_i$，表示系统中的数据段总数。在基于热度的缓存策略中，系统需要记录 Z 个数据段的访问次数，

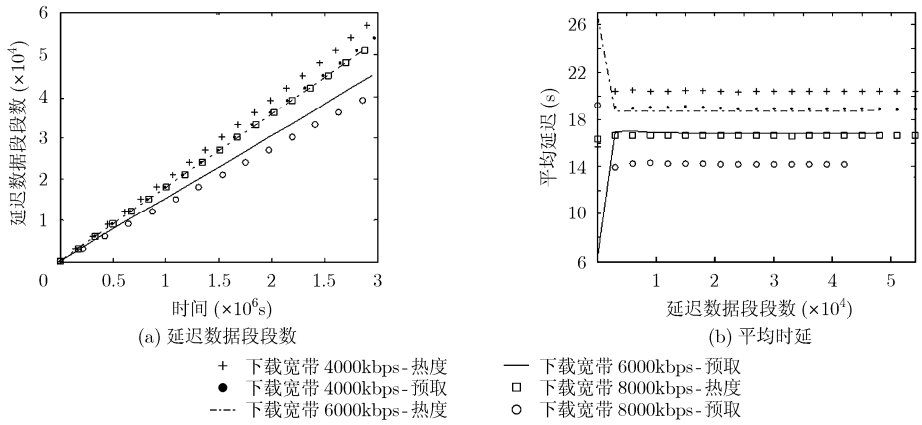

图 3 不同下载带宽的延迟数据段段数和平均延迟

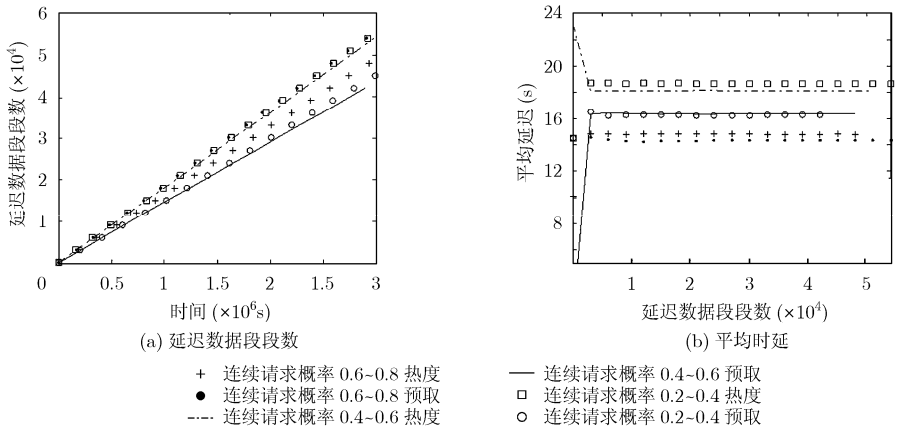

图 4 连续请求概率对数据段延迟的影响

并计算其热度进行排序。而基于数据预取的缓存策略中，由于涉及 VCR 操作的统计，需要记录 Z^2 个数据，然后计算 Z 个数据段的 β 值并排序。可见，基于数据预取的缓存策略需要记录的数据为基于热度的缓存策略的平方，而它们都只需对 Z 个数值进行排序。

5 结束语

本文基于预取带宽分配的方法，对流媒体服务系统中的缓存管理问题进行了研究，提出了减小用户点播延迟的优化公式，在给出次优解的基础上给出在线逼近最优解的方法，能够更好地用于具有 VCR 功能的流媒体系统。仿真结果说明在 VCR 操作特征明显的流媒体系统中，基于预取的缓存策略充分利用系统的带宽以及缓存空间能够有效地降低请求数据段发生延迟的概率，数据段的平均延迟也得到了降低，从而提高用户的点播体验。

参 考 文 献

[1] Shim J, Scheuermann P, and Vingralek R. Proxy cache algorithms: design, implementation, and performance[J]. *IEEE Transactions on Knowledge and Data Engineering*, 1999, 11(4): 549–562.

[2] Liu Jiang-chuan and Xu Jian-liang. Proxy caching for media streaming over the Internet[J]. *IEEE Communications Magazine*, 2004, 42(8): 88–94.

[3] Liang Wei-fang, Huang Ji-hai, and Huang Jian-hua. A distributed cache management model for P2P VoD system[C]. International Conference on Computer Science and Software Engineering, Wuhan, China, Dec. 12–14, 2008: 5–8.

[4] Jiang Wen-bin, Huang Chong, Jin Hai, and Liao Xiao-fei. A new proxy scheme for large-scale P2P VoD system[C]. IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, Shanghai, China, Dec. 17–20, 2008: 512–518.

[5] Alan TS Ip, Liu Jiang-chuan, and John Chi-shing Lui. COPACC: an architecture of cooperative proxy-client caching system for on-demand media streaming[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2007, 18(1): 70–83.

[6] Wu Kun-lung, Yu P S, and Wolf J L. Segmentation of multimedia streams for proxy caching[J]. *IEEE Transactions on Multimedia*, 2004, 6(5): 770–780.

[7] Hyung Rai Oh and Hwangjun Song. Metafile-based scalable caching and dynamic replacing algorithms for multiple videos

over quality-of-service networks[J]. *IEEE Transactions on Multimedia*, 2007, 9(7): 1535–1542.

[8] Chen Song-qing, Shen Bo, Susie Wee, and Zhang Xiao-dong. Segment-based streaming media proxy: modeling and optimization[J]. *IEEE Transactions on Multimedia*, 2006, 8(2): 243–256.

[9] Wang J Z and Yu P S. Fragmental proxy caching for streaming multimedia objects[J]. *IEEE Transactions on Multimedia*, 2007, 9(1): 147–156.

[10] Liu Jie, Liu Yi-na, Cheng Ling-ling, and Tao Jun-cai. Peer caching algorithm based on global segment popularity for P2P VoD system[C]. World Congress on Computer Science and Information Engineering, Los Angeles, USA, Mar. 31–Apr. 2, 2009: 140–144.

[11] Tu Wei. Eckehard Steinbach, Muhammad Muhammad, and Li Xiao-ling. Proxy caching for video-on-demand using flexible starting point selection[J]. *IEEE Transactions on Multimedia*, 2009, 11(4): 716–729.

[12] He Yi-feng, Shen Guo-bin, Xiong Yong-qiang, and Guan Ling. Optimal prefetching scheme in P2P VoD applications with guided seeks[J]. *IEEE Transactions on Multimedia*, 2009, 11(1): 138–151.

巫旭敏：　男，1985 年生，博士生，研究方向为流媒体服务网络.

殷保群：　男，1962 年生，教授，博士生导师，研究方向为网络建模以及网络优化.

黄　静：　男，1985 年生，博士生，研究方向为复杂网络建模与优化.